Mathematical Problems of Computer Science 64, 66–75, 2025.

doi:10.51408/1963-0141

UDC 004.8:004.72

Infrequent Synchronization in Distributed AdaBoost

Arthur K. Oghlukyan[1] and Luis Fernando de Mingo López[2]

[1]Institute for Informatics and Automation Problems of NAS RA, Yerevan, Armenia
[2]Polytechnic University of Madrid, Ispain, Madrid
e-mail: artur.oghlukyan@edu.isec.am, fernando.demingo@upm.es

Abstract

Distributed machine learning has become increasingly vital as data sources continue to expand geographically. Traditional ensemble methods such as AdaBoost demonstrate impressive predictive capabilities but often require frequent synchronization across nodes, resulting in significant communication overhead. This paper introduces a novel paradigm of **infrequent synchronization** in which nodes perform multiple rounds of local AdaBoost before exchanging partial or complete model updates. The potential advantages include reduced communication costs, the ability to handle intermittent connectivity, and competitive accuracy compared to fully synchronized approaches. A real-world use case in the trucking industry is presented to demonstrate the feasibility and value of this new approach. The paper concludes by outlining future directions and the expected impact on communication-efficient distributed learning.

Keywords: Distributed AdaBoost, Infrequent Synchronization, Ensemble Learning, Communication-Efficient Learning, Federated Boosting, Weak Learners, Scalability, Fault Tolerance, Real-World Deployment.

Article info: Received 16 May 2025; sent for review 5 June 2025; accepted 7 October 2025.

Acknowledgments: The authors wish to thank contributors from various open-source distributed ML projects for their insights into scalable ensemble frameworks.

1. Introduction

1.1 Background

The evolution of big data analytics has necessitated the adoption of distributed machine learning frameworks that can operate across geographically dispersed nodes. Ensemble algorithms, particularly AdaBoost [1], stand out for their ability to transform weak learners into a strong classifier through iterative reweighting of training examples. Yet, the classical distributed deployment of AdaBoost relies on frequent data or model exchanges, often every boosting round to maintain a coherent global model [2]. This can be problematic in scenarios where communication is costly, bandwidth is limited, or connectivity is intermittent [3].

1.2 Problem Statement

This research explores a delayed or infrequent synchronization strategy in distributed AdaBoost to minimize communication overhead without substantially sacrificing model accuracy [4]. Specifically, it examines whether AdaBoost's iterative process can be adapted to function effectively under limited exchange conditions, an area partially explored in [5, 6] but remaining relatively undeveloped for classical boosting approaches.

1.3 Research Objectives

1. **Reduced Communication Overhead**: Demonstrate how the frequency of synchronization rounds can be lowered to a fraction of the total boosting iterations while still retaining high model accuracy.
2. **Adaptive Local Training**: Investigate how local reweighting schemes can operate in isolation for multiple iterations to mitigate the stale model challenge.
3. **Real-World Feasibility**: Illustrate a use case where intermittent connectivity is common, namely the trucking industry, to validate the proposed method's applicability and benefits.

2. Related Work

2.1 Communication-Efficient Ensemble Methods

Studies in communication-efficient boosting [7] and local update strategies [8] establish that sparse or delayed parameter sharing can preserve accuracy under theoretical guarantees. Distributed boosting algorithms such as PreWeak and AdaSampling reduce synchronization overhead by transmitting partial updates or sampled data to a central coordinator [9]. Nonetheless, these methods often necessitate specialized sampling or model pruning to ensure convergence.

2.2 Federated Learning Paradigms

Federated learning (FL) promotes infrequent synchronization, a property particularly valuable in privacy-sensitive industries such as healthcare and finance [10]. Techniques like Federated Averaging (FedAvg) allow clients to perform multiple local gradient-descent updates before transmitting aggregated model parameters to a central coordinator, thereby substantially reducing communication overhead [5]. Although the underlying motivation of decreasing communication is shared, FedAvg and distributed AdaBoost differ fundamentally in their learning dynamics. FedAvg operates on parametric, differentiable models whose parameters can be averaged meaningfully across clients. In contrast, distributed AdaBoost aggregates weak learners through weighted voting rather than parameter averaging, and its communication steps revolve around synchronizing model weights, error rates, or classifier outputs rather than gradient-based updates. This distinction highlights that, while both paradigms benefit from reduced communication frequency, the mechanisms enabling synchronization in distributed AdaBoost require algorithmspecific coordination rather than straightforward parameter averaging, motivating the development of tailored synchronization strategies.

2.3 Distributed Systems and AdaBoost Synchronization

Distributed systems rely on principles such as scalability, consistency, and fault tolerance [11]. Achieving these principles often involves trade-offs; scalable systems commonly sacrifice strict consistency to ensure fault tolerance and availability [12]. Ensemble methods like AdaBoost face unique synchronization challenges due to their sequential nature; each iteration requires updated global error rates to adjust sample weights, typically necessitating synchronous communication [1, 13].

In contrast to federated learning's flexibility in synchronization (e.g., asynchronous and hybrid approaches such as FedBuff), distributed AdaBoost methods commonly enforce synchronous communication rounds [14]. Empirical analyses, such as LoAdaBoost FedAvg, illustrate AdaBoost's integration within federated frameworks by combining synchronous aggregation with adaptive weighting strategies to enhance accuracy and efficiency [15]. However, AdaBoost's inherent sequential dependencies limit the potential for asynchronous updates, reinforcing the necessity of carefully managed synchronization mechanisms [16].

3. Proposed Methodology

The primary goal is to minimize synchronization events in a distributed AdaBoost framework while still preserving sufficient model accuracy. Each node locally runs multiple rounds of AdaBoost on its partition of data, updating instance weights and training weak learners without requiring continuous global communication [7, 5]. Periodically, but as rarely as feasible, these partial updates are exchanged and merged into a global ensemble, aligning node-specific reweighting strategies and capturing the collective knowledge [6]. This strategy builds upon the classical boosting concept [1] yet limits the overhead typically associated with every-round synchronization. While more local iterations between communications can slightly increase the risk of model drift, careful selection of when and how often to synchronize helps balance reduced bandwidth usage with acceptable predictive performance in large-scale or bandwidth-constrained environments.

3.1 Algorithmic Framework

The novel contribution is an **infrequent synchronization strategy** for AdaBoost. Instead of synchronizing after each boosting iteration, each node trains locally for K rounds, storing partially updated models in a local buffer. After these K rounds, the node communicates with a central server or peer nodes to merge and refine the global ensemble. The process repeats until the desired number of boosting rounds is reached.

Key Insight: By decoupling local updates, the model allows partial divergence in node-specific weight distributions. Periodic global synchronization steps mitigate error accumulation and realign the ensemble.

3.2 Pseudocode

```
Algorithm: InfrequentSyncAdaBoost

Input:
   D = {D_1, D_2, ..., D_N}  // Partition of data across N nodes
   T              // Total number of boosting rounds
   K              // Frequency of synchronization
   base_learner     // Base classifier for AdaBoost

Initialize:
   For each node n:
      Initialize local sample weights w_n(i) = 1 / |D_n|

For round t = 1 to T:
   For each node n in parallel:
      1. Train weak classifier h_n(t) on D_n using current weights w_n
      2. Compute error_n(t) = Σ [w_n(i) * I(h_n(t)(x_i) ≠ y_i)]
      3. Compute α_n(t) = 0.5 * ln((1 - error_n(t)) / error_n(t))
      4. Update weights:
        w_n(i) = w_n(i) * exp(-α_n(t) * y_i * h_n(t)(x_i))
      5. Normalize w_n(i)

   If t % K == 0 or t == T:
      // Synchronize across nodes
      6. Gather {h_n(t'), α_n(t')} for t' in {t-K+1, ..., t} from each node
      7. Construct global ensemble H_g(r) by merging or aggregating these weak classifiers
      8. (Optional) Prune or refine ensemble if it becomes too large
      9. Broadcast H_g(r) to all nodes, so they can align partial states

Output:
   Final aggregated ensemble: H_g(T)
```

3.3 Complexity Analysis

- **Communication**: Reduced from O(NT) in classical distributed AdaBoost (where each node synchronizes every iteration) to O(NT / K) in the proposed infrequent scheme.
- **Computation**: Minimal additional overhead arises from merging partial ensembles, which can be done through a simple central aggregator. Local computations remain identical to standard AdaBoost.
- **Potential Trade-offs**: Longer local training phases might introduce greater divergence from the global optimum, demanding careful parameter tuning (e.g., selection of K).

4. Evaluation

4.1 Experimental Setup

Three variants of distributed AdaBoost are compared:

1. **Fully Synchronized (Baseline)**: Synchronization at every boosting round.
2. **Moderately Synchronized**: Synchronization every 5 rounds.
3. **Infrequent Synchronization**: Synchronization only once or twice (e.g., every 25 or 50 rounds).

A sample result table and visualizations based on the evaluation metrics for 100 boosting rounds is shown in Table 1.

Table 1. Synchronization analysis

Sync Strategy	Sync Rounds	Communication (MB)	Accuracy (%)	TrainingTime (s)
Fully Synchronized	100	120.0	95.4	240
Every 5 Rounds	20	24.0	94.9	180
Every 25 Rounds	4	4.8	94.2	160
Once (at End)	1	1.2	91.5	140

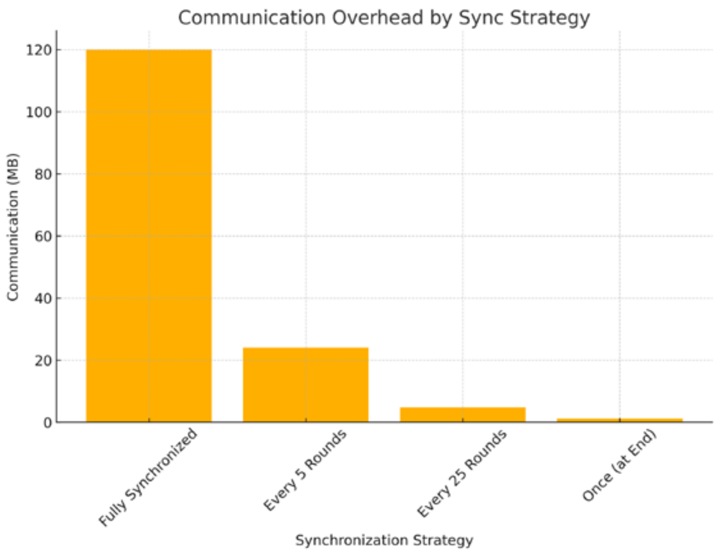

Fig. 1. Communication overhead by synchronization strategy.

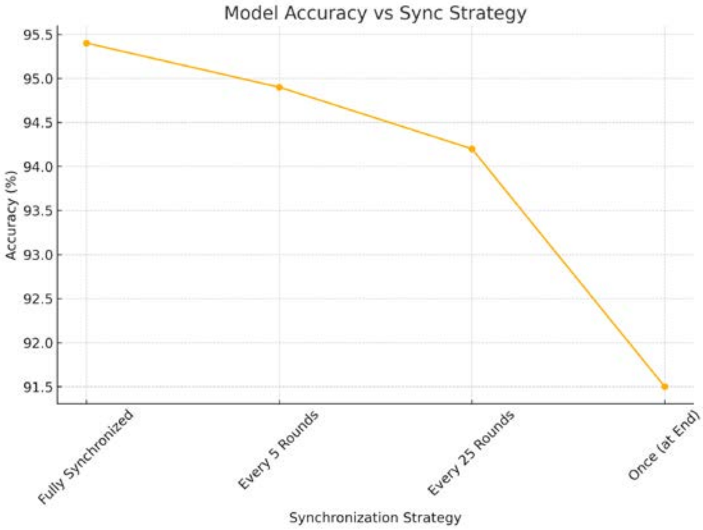

Fig. 2. Model accuracy vs Synchronization strategy.

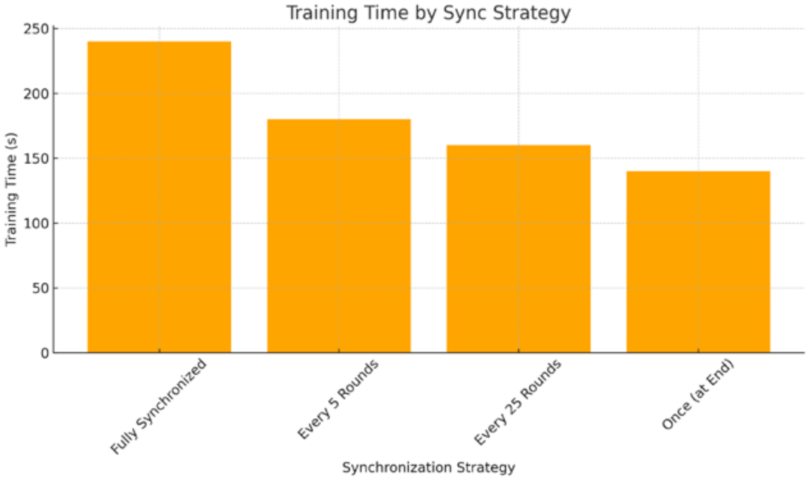

Fig. 3. Training time by synchronization strategy.

Although the final accuracy for the "Once (at End)" synchronization strategy is lower at 91.5% compared to more frequent synchronization, it remains sufficiently high for many real-world applications. In scenarios where faster training time, reduced communication overhead, or intermittent connectivity takes priority, a slight reduction in accuracy can be acceptable. This performance level still represents a robust predictive capability, ensuring that the trade-off between model quality and limited synchronization is justifiable in numerous large-scale or bandwidth-constrained deployments.

4.2 Real-World Use Case: Trucking Industry

4.2.1 Context and Data Generation

Each truck in a large fleet is equipped with sensors gathering telematics data: fuel usage, engine temperature, speed, brake usage, GPS location, etc. Due to remote driving routes and sporadic connectivity, the trucks can only synchronize with the central office every few hours or at designated checkpoints.

4.2.2 Application

- **Predictive Maintenance**: Early detection of mechanical issues based on aggregated sensor data.
- **Fuel Efficiency Optimization**: Identifying fuel-wasting driving behaviors across different terrains.
- **Driver Safety Analysis**: Monitoring and alerting high-risk driving patterns (sharp braking, speeding).

4.2.3 Implementation

Local AdaBoost runs on each truck for multiple iterations, updating the distribution of "tricky" instances found in that truck's routes. When connectivity allows, all partial models are transmitted to a central aggregator, which merges them and redistributes global updates.

Outcome: Even with one or two synchronizations per day, fleet-wide predictive performance remains competitive, enabling near-real-time insights into vehicle health and driver habits without overwhelming the limited communication infrastructure.

5. Conclusion and Future Work

The proposed **infrequent synchronization strategy** for distributed AdaBoost addresses pressing challenges in large-scale, communication-constrained settings. Preliminary analyses indicate that local training for multiple rounds before synchronizing can significantly reduce communication overhead with a modest trade-off in convergence speed or final accuracy. Future work will involve formalizing convergence bounds, exploring adaptive synchronization schedules based on node-specific performance, and implementing privacy-preserving protocols to handle sensitive data (e.g., driver habits or proprietary operational metrics).

References

[1] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[2] Y. Zhang and J. Huan, “Inductive multi-task learning with multiple view data”, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 543–551, 2011.

[3] L.-M. Ang, K. P. Seng et al., “Big sensor data systems for smart cities”, *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 468–476, 2018.

[4] J. Hamer, G. Quinlan and S. Marshland, “FedBoost: Communication-efficient federated learning with boosting”, *Proceedings of the International Conference on Distributed Machine Learning*, pp. 1–8, 2020.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data”, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (AISTATS), pp. 1273–1282, 2017.

[6] K. Cheng, T. Fan, Y. Jin et al., “SecureBoost: A lossless federated learning framework”, *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 776–788, 2021.

[7] M.-F. Balcan, A. Blum and S. Fine, “Communication efficient distributed learning”, *Proceedings of the 25th Annual Conference on Learning Theory*, pp. 35.1–35.22, 2012.

[8] T. Li, A. K. Sahu, M. Zaheer et al., “Federated optimization in heterogeneous networks”, *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[9] Y.-Y. Chiang, C.-J. Hsieh and I. S. Dhillon, “Communication-efficient distributed boosting algorithms”, *Proceedings of the International Conference on Machine Learning* (ICML), pp. 139–147, 2014.

[10] P. Kairouz, H. B. McMahan et al., “Advances and open problems in federated learning”, *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[11] S. Gilbert and N. Lynch, “Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services”, *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002.

[12] M. Li, D. G. Andersen, A. J. Smola and K. Yu, “Communication efficient distributed machine learning with the parameter server”, *Advances in Neural Information Processing Systems*, vol. 27, pp. 19–27, 2014.

[13] A. Karbasi and K. G. Larsen, “Parallel boosting algorithms: Limitations and possibilities”, *Journal of Machine Learning Research*, vol. 25, pp. 1–34, 2024.

[14] Y. Fraboni, R. Vidal, L. Kameni and M. Lorenzi, “FedBuff: Asynchronous federated learning with buffered updates”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 102–114, 2023.

[15] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng and D. Liu, “Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records”, *Journal of Biomedical Informatics*, vol. 99, p. 103291, 2020.

[16] M. Polato, A. Sperduti and F. Chierichetti, “AdaBoost.F: Federated boosting with decision trees”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2485–2496, 2023.

Սահմանափակ համաժամեցում բաշխված AdaBoost-ում

Արթուր Կ. Օղլուկյան[1] և Լուիս Ֆերնանդո դե Մինգո Լոպես[2]

[1]ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ, Երևան, Հայաստան
[2] Մադրիդի պոլիտեխնիկական համալսարան, Իսպանիա, Մադրիդ
e-mail: artur.oghlukyan@edu.isec.am, fernando.demingo@upm.es

Ամփոփում

Բաշխված մեքենայական ուսուցումը դարձել է ավելի ու ավելի կարևոր, քանի որ տվյալների աղբյուրները շարունակում են ընդլայնվել աշխարհագրորեն: Ավանդական անսամբլային մեթոդները, ինչպիսին է AdaBoost-ը, ցուցադրում են տպավորիչ կանխատեսողական հնարավորություններ, բայց հաճախ պահանջում են հաճախակի համաժամեցում հանգույցների միջև, ինչը հանգեցնում է հաղորդակցման զգալի ծանրաբեռնվածության: Այս առաջարկը ներկայացնում է հազվադեպ համաժամեցման նոր մոդել, որի դեպքում տեղական AdaBoost-ի բազմաթիվ փուլեր են իրականացվում մոդելի մասնակի կամ ամբողջական թարմացումների փոխանակումից առաջ: Հնարավոր առավելություններից են հաղորդակցման ծախսերի կրճատումը, ընդհատվող կապը կառավարելու հնարավորությունը և մրցակցային ճշգրտությունը՝ համեմատած լիովին համաժամեցված մոտեցումների հետ: Ներկայացվում է բեռնափոխադրումների ոլորտում իրական աշխարհի օգտագործման դեպք՝ այս նոր մոտեցման իրագործելիությունն ու արժեքը ցույց տալու համար: Հոդվածը եզրափակվում է ապագա ուղղությունների և հաղորդակցման արդյունավետ բաշխված ուսուցման վրա սպասվող ազդեցության ուրվագծմամբ:

Բանալի բառեր՝ բաշխված AdaBoost, հազվադեպ սինխրոնիզացիա, համալիր ուսուցում, հաղորդակցման արդյունավետ ուսուցում, ֆեդերատիվ ուժեղացում, թույլ սովորողներ, մասշտաբայնություն, սխալների հանդուրժողականություն, իրական աշխարհի տեղակայում

Нечастая синхронизация в распределенном AdaBoost

Артур К. Оглукян и Луис Фернандо де Минго Лопес

[1]Институт проблем информатики и автоматизации НАН РА, Ереван, Армения
[2]Политехнический университет Мадрида, Испания, Мадрид
e-mail: artur.oghlukyan@edu.isec.am, fernando.demingo@upm.es

Аннотация

Распределенное машинное обучение становится все более важным, поскольку источники данных продолжают расширяться географически. Традиционные ансамблевые методы распознавания, такие как AdaBoost, демонстрируют впечатляющие возможности прогнозирования, но часто требуют активной синхронизации между узлами, что приводит к значительным накладным расходам на связь. Это предложение вводит новую парадигму нечастой синхронизации, в которой несколько раундов локального AdaBoost выполняются до обмена частичными или полными обновлениями модели. Потенциальные преимущества включают снижение затрат на связь, способность обрабатывать прерывистое подключение и конкурентоспособную точность по сравнению с полностью синхронизированными подходами. Представлен реальный пример использования в отрасли грузоперевозок, чтобы продемонстрировать осуществимость и ценность этого нового подхода. Статья завершается описанием будущих направлений и ожидаемого влияния на эффективное с точки зрения связи распределенное обучение.

Ключевые слова: распределённый AdaBoost, нечастая синхронизация, ансамблевое обучение, коммуникационно-эффективное обучение, федеративное обучение, слабые обучающиеся, масштабируемость, отказоустойчивость, развертывание в реальных условиях.